NMDS ANALYIS OF SEQUENCING DATA

Marcus Johnson
marcus.johnson@ec.gc.ca

Created: January 15, 2024
Updated: January 16, 2024

BACKGROUND

Non-metric multidimensional scaling, or NMDS, is a subset of statistical tools known as principal component analysis (PCA). Both NMDS and PCA fall under the umbrella of multivariate analysis, aiming to reveal patterns and relationships within complex datasets. PCA is a general technique used to transform multiple measurements of a sample's variables into a set of linearly uncorrelated variables known as principal components. These components capture the maximum variance in the data, providing a concise representation of the original information. PCA is versatile and applicable to diverse types of data, making it a valuable tool in exploring the underlying structure of complex datasets.

NMDS, as a specialized technique within the PCA family, extends the capabilities of traditional PCA by incorporating rank orders and ordinal relationships into its analysis. Unlike PCA, which preserves metric properties, NMDS is particularly well-suited for situations where the dissimilarity measures are non-metric or ordinal in nature. In ecological studies, for instance, taxonomic information often involves rankings rather than precise measurements, making NMDS an ideal choice for exploring and visualizing patterns of similarity or dissimilarity among samples.

The significance of NMDS becomes apparent when dealing with datasets exhibiting non-linear relationships or when the nature of dissimilarity measures requires a more flexible approach. For example, when analyzing ecological communities, the presence or absence of species may be more accurately captured by rank orders, reflecting the relative importance of different species within a community.

CONTENTS

1. PREREQUISITES

1.1 Software and required packages

This tutorial of NMDS analysis and plotting assumes you have a basic understanding of R. For guidance on installing R and its dependencies, refer to the manual "Installation and set up of R and its affiliated packages."

This analysis requires the use of the **vegan** package.[1] **vegan** should be installed by default with the base version of R, but can be installed using the given commands. The tutorial will also use functions in the **tidyverse** package, including **ggplot2**.

```
install.packages("vegan")
library(vegan)
library(tidyverse)
```

1.2 Data format

The following analysis assumes you are working with data containing normalised sequencing counts. For guidance on normalising your data, refer to the tutorial "Data normalisation using DEseq2."

The following analysis requires two data files:

1. A counts table. The counts table should be formatted such that the sequences or sequence identifiers are organized by row. That is, the first column of the data set identifies the count object (*i.e.* base 16 hash identifier, OUT, ASV, sequence, etc.) and the rest of the columns are formatted with the sample ID at the top. An example is presented here:

	sample01	sample02	sample03	sample04	sample05	sample06
240acf5d	86	80	11	53	11	77
df4a782c	43	13	42	83	22	74
536bc65f	34	47	57	85	98	81

2. A metadata table. The metadata table will include the sample IDs listed by row and each test variable for the sample. While certain test information may be contained in the sample name (*e.g.* treatment 2, replicate 3), it is best practice to include all the information about a given sample in the metadata table. A sample is presented here:

SampleID	**Treatment**	**Replicate**	**ContaminantConc**	**ControlGroup**
sample01	T01	R1	0	TRUE

Note that R cannot handle column names with spaces or complex characters; R will replace these with a period (.). It is best practice to not use complex characters in your sample IDs to prevent unexpected errors when searching for samples or matching information according to your metadata file. Examples presented here demonstrate good practice naming conventions.

1.3 Included example

An example is included to demonstrate the following principles and can be followed along. The example takes OUT count values and sample information from Caporaso et al. (2011).[2] The example imports tab-separated text files "`.txt`" but other versions of data may require different ways of importing data.

```
df_otu <- read.table(file = "example_data/counts.txt",
                     sep = "\t", header = T, row.names = 1)
df_meta <- read.table(file = "example_data/metadata.txt",
                      sep = "\t", header = T, row.names = 1)
```

We can preview the objects by using the `head()` function, as seen previously.

```
head(df_meta)
        SampleID Description    Location                                    Notes
CL3          CL3        Soil Environment   Calhoun South Carolina Pine soil, p…
CC1          CC1        Soil Environment   Cedar Creek Minnesota, grassland, p…
SV1          SV1        Soil Environment Sevilleta new Mexico, desert scrub, p…
M31Fcsw  M31Fcsw       Feces       Human    M3, Day 1, fecal swab, whole body …
M11Fcsw  M11Fcsw       Feces       Human   M1, Day 1, fecal swab, whole body s…
M31Plmr  M31Plmr        Skin       Human    M3, Day 1, right palm, whole body …
```

2. ANALYSIS

2.1 Correctly formatting data

The functions used to calculate and present an NMDS plot rely on the type of data imported to be correct. Objects in R, like all programming languages, are of a certain type of class. The class of any object can be determined with the `class()` function. Specific columns of a data frame can be selected using the $ operator.

```
# check data type
class(df_meta$Description)
[1] "character"
class(df_meta$Location)
[1] "character"
```

The two columns of the metadata data frame that we wish to use for our analysis were imported as character strings. However, the NMDS analysis requires that these be treated as a factor. A factor is a categorical variable that represents a fixed set of values These can be unordered (like locations) or order (like nominal concentrations).

To convert character strings in your metadata data frame to factors, you can use the `as.factor()` function in R. Here's how you can do it for specific columns in your metadata data frame:

```
# convert test variables to factors
df_meta$Location <- as.factor(df_meta$Location)
df_meta$Description <- as.factor(df_meta$Description)
```

In the event that factors need to be ordered, say in the example of increasing nominal concentrations, this can be completed using the **`ordered()`** function:

```
# example of an ordered list
location_order <- c("Human", "Environment", "Mock") # list out all factors
ordered(df_meta$Location, location_order)
 [1] Environment Environment Environment Human       Human
 [6] Human       Human       Human       Human       Human
[11] Environment Environment Environment Environment Environment
[16] Environment Environment Environment Environment Environment
[21] Environment Human       Human       Mock        Mock
[26] Mock
Levels: Human < Environment < Mock
```

Once the data has been correctly formatted, the analysis can proceed.

2.2 Conducting analysis

The NMDS analysis begins by calculating a dissimilarity indices using the **`vegdist()`** function. The dissimilarity indices accepts numerous methods for conducting the calculation; the one we will use here is the Bray-Curtis analysis, but all the options can be viewed using the **`?vegdist`** command.

Then an analysis of variance is calculated using the distance matrix using the **`adonis2()`** function. The function requires you to set a formula based on the way the study was designed.

```
# Bray-Curtis analysis
bray_curtis_dist <- vegdist(t(df_otu), method = "bray")

# conduct ANOVA using dist. matrix
permanova_result <- adonis2(
  formula = bray_curtis_dist ~ Description,          # set formula of variables
  data = df_meta)                    # include the relevant metadata data frame
```

In the above example, the test samples are separated only based on Description, because the other factor Location is also described by Description and is therefore not independent. However, an example is presented here for use if multiple factors are required for the analysis. Notice the formula is set using the + operator. Formulas in R follow the general style `y ~ x1 + x2`.

```
# example ANOVA with more than one variable
example_permanova <- adonis2(
  formula = bray_curtis_dist ~ Description + Location,   # formula of variables
  data = df_meta)                    # include the relevant metadata data frame
```

The results from the permanova can be previewed using the **print()** function.

```
print(permanova_result) # collect results
Permutation test for adonis under reduced model
Terms added sequentially (first to last)
Permutation: free
Number of permutations: 999

adonis2(formula = bray_curtis_dist ~ Description, data = df_meta)
            Df SumOfSqs      R2      F Pr(>F)
Description  8   7.7744 0.67562 4.4259  0.001 ***
Residual    17   3.7327 0.32438
Total       25  11.5070 1.00000
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The significance of the model is given by the F statistic, above demonstrated at > 0.001, telling us that the Description factor played a significant role in the explaining the differences between groups.

The NMDS can be conducted using the function **metaMDS()** on the **bray_curtis_dist** object previously created.[1]

```
# conduct an NMDS on the Bray-Curtis matrix
nmds_result <- metaMDS(bray_curtis_dist)
Run 0 stress 0.1770946
Run 1 stress 0.1888321
(Run 2-19 omitted)
Run 20 stress 0.1710613
... New best solution
... Procrustes: rmse 3.377271e-05  max resid 9.289718e-05
... Similar to previous best
*** Best solution repeated 1 times

Call:
metaMDS(comm = bray_curtis_dist)

global Multidimensional Scaling using monoMDS

Data:     bray_curtis_dist
Distance: bray

Dimensions: 2
Stress:     0.1710613
Stress type 1, weak ties
Best solution was repeated 1 time in 20 tries
The best solution was from try 20 (random start)
Scaling: centring, PC rotation, halfchange scaling
Species: scores missing
```

[1] Note that in the results, certain lines of output have been omitted for brevity.

What the NMDS is doing in each step is iteratively generating best principal component and tries to find a stable solution using several random starts. In addition, it standardizes the scaling in the result, so that the configurations are easier to interpret, and is capable of adding species scores to the site ordination, although that was not conducted here.

The scores from the NMDS analysis can be exported into a data frame so that it may be plotted using R's data visualisation capabilities. At this stage, we will reincorporate information contained in the metadata data frame (using the **`cbind()`** function).

```
# collect the scores from the nmds
nmds_data <- as.data.frame(scores(nmds_result))
nmds_data <- cbind(nmds_data, df_meta) # bind the columns from the meta data
```

3. GRAPHING THE RESULTS

3.1 Generating graphs

The resulting NMDS information can be presented using the **`tidyverse`** package **`ggplot2`**.

```
# create ggplot
nmds_plot <- ggplot(data = nmds_data, aes(x = NMDS1, y = NMDS2)) +
  geom_point(aes(col = Description, shape = Location), size = 2) +
  theme_bw()

nmds_plot # view the plot
```

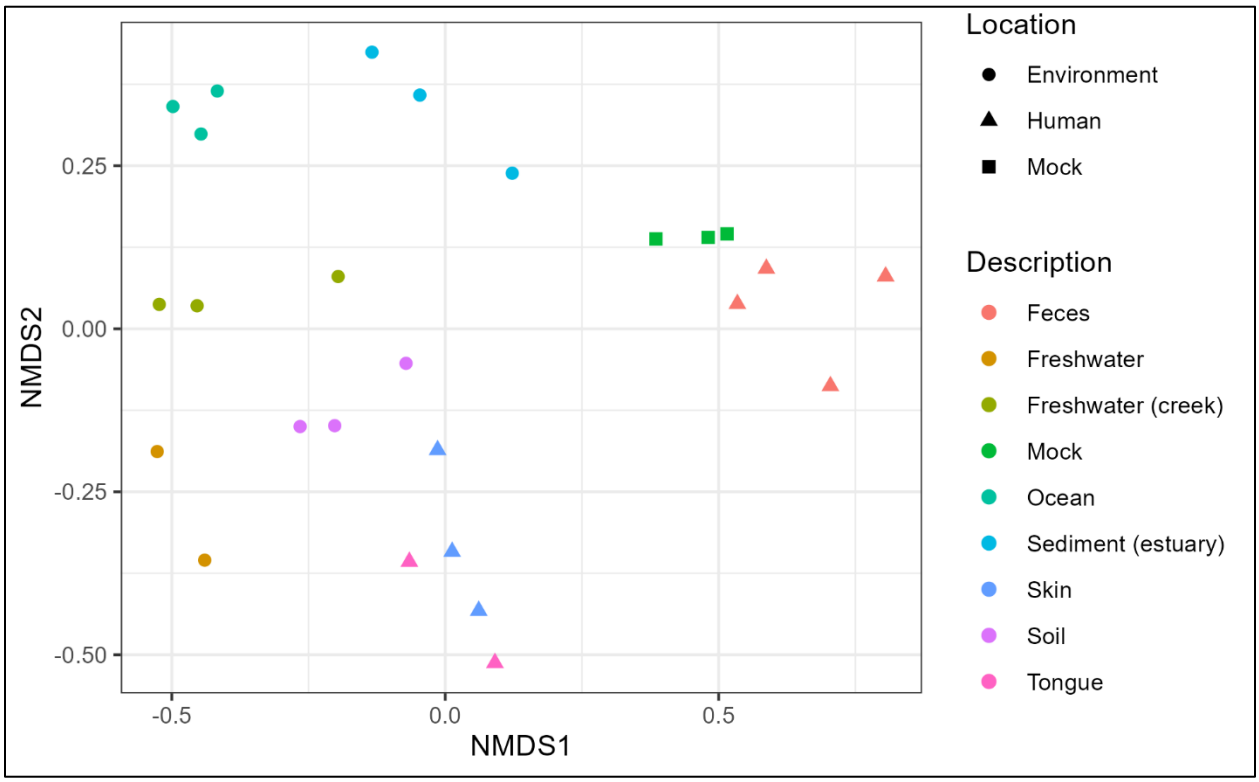

Figure 1: Resulting NMDS plot.

The graph can be generated using the different factors specified in *§ 2.1 Correctly formatting data*. Numerical and non-factor variables will be treated differently from factored variables. More information about graphing can be found in the tutorial "Graphing in R using ggplot2."

3.2 Exporting graphs

Figures created by `ggplot` can be exported using the `ggsave()` function.

```
ggsave(
  filename = "output/NMDS_scatterplot.PNG", # within the folder "output"
  plot = nmds_plot,
  width = 6.5, height = 5,
  dpi = 300)
```

The above `ggsave()` function generate a "`.PNG`" filetype image in the folder `output`.

BIBLIOGRAPHY

(1) Oksanen, J.; Simpson, G. L.; Blanchet, F. G.; Kindt, R.; Legendre, P.; Minchin, P. R.; O'Hara, R. B.; Solymos, P.; Stevens, M. H. H.; Szoecs, E.; Wagner, H.; Barbour, M.; Bedward, M.; Bolker, B.; Borcard, D.; Carvalho, G.; Chirico, M.; Caceres, M. D.; Durand, S.; Evangelista, H. B. A.; FitzJohn, R.; Friendly, M.; Furneaux, B.; Hannigan, G.; Hill, M. O.; Lahti, L.; McGlinn, D.; Ouellette, M.-H.; Cunha, E. R.; Smith, T.; Stier, A.; Braak, C. J. F. T.; Weedon, J. Vegan: Community Ecology Package, 2022. https://cran.r-project.org/web/packages/vegan/index.html.

(2) Caporaso, J. G.; Lauber, C. L.; Walters, W. A.; Berg-Lyons, D.; Lozupone, C. A.; Turnbaugh, P. J.; Fierer, N.; Knight, R. Global Patterns of 16S RRNA Diversity at a Depth of Millions of Sequences per Sample. *Proceedings of the National Academy of Sciences* **2011**, *108* (supplement_1), 4516–4522. https://doi.org/10.1073/pnas.1000080107.